

Instrukcja 1

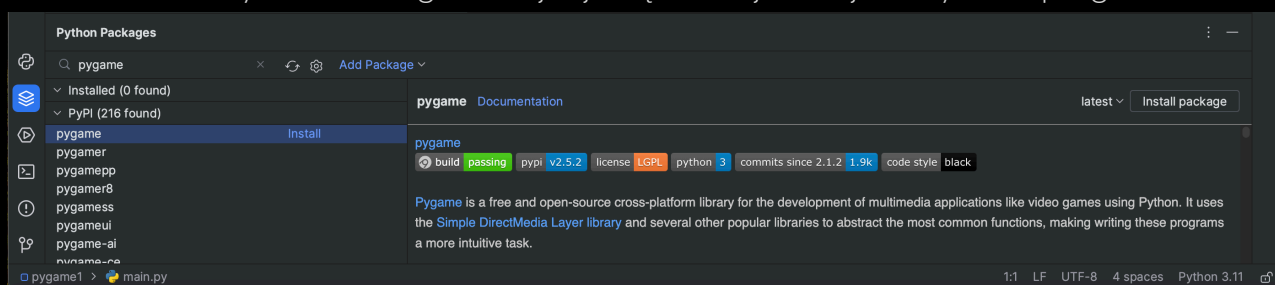
Wprowadzenie do biblioteki Pygame

Podstawy tworzenia gier

Instalacja biblioteki

Pygame jest zewnętrzną biblioteką kodu (zbiór przygotowanych wcześniej funkcji i narzędzi, które programiści mogą wykorzystać w swoim kodzie, bez konieczności tworzenia ich „od zera”). Z tego powodu nie jest ona dołączona do standardowej instalacji Pythona. Istnieją dwa sposoby instalacji bibliotek:

1. Wykorzystując menadżera instalacji pakietów pip
`pip install pygame`
2. Instalując bibliotekę poprzez menadżera pakietów programu PyCharm
Menadżer „Python Packages” znajduje się z lewej dolnej strony okna programu



Podstawy gry

1. Na początku konieczne jest zaimportowanie biblioteki do projektu
`import pygame`
2. Następnie biblioteka musi zostać zainicjalizowana
`pygame.init()`
3. W kolejnym kroku należy utworzyć okno gry. W tym celu konieczne jest sprecyzowanie wymiarów okna, jego sposobu wyświetlenia oraz podpisu
`width = 500`
`height = 500`
`game_view = pygame.display.set_mode((width, height))`
`pygame.display.set_caption(„Pierwsza gra w Pygame”)`

- Następnie konieczne jest utworzenie „Pętli gry” - powtarzany w nieskończoność fragment kodu, który kontroluje przebieg działania całej gry. Będzie się on zajmował aktualizacją logiki gry, rysowaniem grafiki, obsługą interakcji gracza oraz zdarzeń. Pętla będzie działać do momentu, gdy użytkownik postanowi zakończyć grę.

```
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    pygame.display.update()
pygame.quit()
```

- Po uruchomieniu programu wyświetlone zostanie puste okno gry.

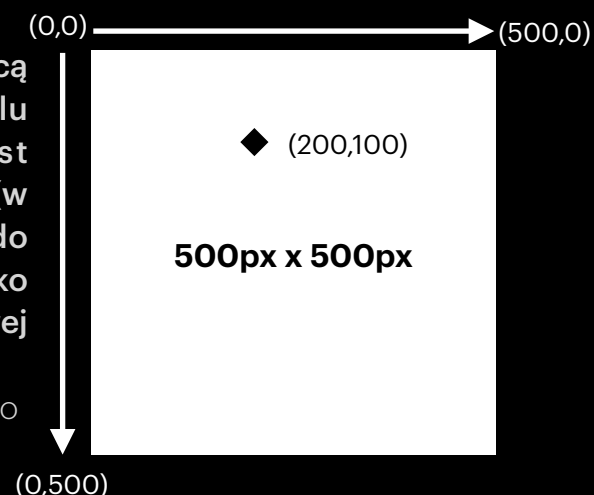
Rysowanie kształtów

- Przed rozpoczęciem rysowania konieczne jest **zadeklarowanie koloru**. Biblioteka Pygame obsługuje deklaracje koloru w formie **nazwy**, kodu **RGB** (mieszanki wartości Red Green Blue między 0-255) oraz na podstawie kodu koloru w formacie **HEX**

```
# Pygame colors
red = pygame.Color("red")
# RGB colors
black = (0, 0, 0)
red = (255, 0, 0)
green = (0, 255, 0)
blue = (0, 0, 255)
white = (255, 255, 255)
# HEX colors
Orange = "#FFA500"
```

- Rysowanie na „Oknie gry” odbywa się za pomocą funkcji należących do `pygame.draw`. W celu narysowania „czegośkolwiek” konieczne jest wskazanie **współrzędnych** punktu początkowego (w pikselach). Sposób obliczenia jest podobny do układu współrzędnych z matematyki, jednak jako osie służą górna i lewa krawędź okna. Po prawej stronie zamieszczono przykład.

Skojarzenie: szerokość to oś liczbowa od lewej do prawej, a wysokość to oś liczbowa od góry do dołu.



3. W celu narysowania **prostokąta** (i **kwadratu**) należy wskazać okno gry, kolor (lub jego zmienną), a następnie współrzędne jego punktu **początkowego** (lewy górny róg prostokąta), **długość** prostokąta i **szerokość** prostokąta.

```
pygame.draw.rect(game_view, "red", (150, 150, 200, 100))
```

Powyższy przykład przedstawia rysowanie prostokąta o długości 200, wysokości 100, a jego punkt początkowy znajduje się w miejscu (150,150)

UWAGA: Jeżeli element ma być „stały” w oknie gry, to należy go umieścić PRZED pętlą gry. Umieszczenie go wewnątrz będzie powodowało ponowne rysowanie elementu przy każdej iteracji pętli!

4. W celu narysowania **koła** należy wskazać okno gry, kolor (lub jego zmienną), a następnie współrzędne punktu **środkowego** oraz **średnicy** koła.

```
pygame.draw.circle(game_view, "green", (250, 250), 50)
```

5. W celu narysowania linii należy wskazać okno gry, kolor (lub jego zmienną), a następnie punkt początkowy, punkt końcowy oraz grubość linii.

```
pygame.draw.line(game_view, "yellow", (100, 400), (400, 400), 5)
```

6. Pozostałe kształty są dostępne w [dokumentacji Pygame](#).

7. Zmianę koloru tła wykonuje funkcja fill

```
game_view.fill("gray")
```

UWAGA: Kolejność rysowania ma znaczenie!

Zadanie podstawowe

Utwórz nowy projekt. Zainstaluj / zaimportuj bibliotekę **Pygame**. Utwórz okno o wysokości **600px**. i długości **800px**. Narysuj **prostokąt**, **kwadrat**, **koło** oraz **linię**. Wykorzystaj **kolory**. Plik **main.py** prześlij na adres szkola@davidkasperek.com

Część rozszerzona znajduje się na kolejnej stronie

Rozszerzenie

Tuple

Tupla ([tuple](#)) jest strukturą danych podobną do listy. Przechowuje [kolekcję](#) elementów w uporządkowany sposób. Tupla w przeciwieństwie do listy nie może być zmieniana - nie ma możliwości dodawania, usuwania lub modyfikowania elementów. Sprawia to, że jest bardziej bezpieczna, ponieważ chroni przed przypadkowymi zmianami. Każdy element tupli jest indeksowany (przechowywany w określonej kolejności), a sposób ich odczytu jest podobny do listy.

```
tuple_example = (1, 2, "three", 4.0)
print(tuple_example[0]) # pierwszy element
```

W poprzednim zadaniu tuple były wykorzystywane do:

- ustawienia rozmiaru okna gry
- kodowania kolorów RGB
- podawania współrzędnych punktów

Zadanie rozszerzone

Wykonaj [zadanie podstawowe](#). Napisz [funkcję](#), która uprości proces rysowania kwadratu: po podaniu w parametrze współrzędnych [punktu początkowego](#), funkcja narysuje [czerwony kwadrat](#) o rozmiarze 50px x 50px we wskazanym miejscu.